

中華電信 工務類:專業職(四) 第一類專員

資訊系統開發及維運

100年 計概 中華電信

結構化程式設計的三種基本組構

1. 循序(sequence)
2. 決策(decision)
3. 重複(repetition)

堆疊(stack)： Last in First out

佇列(queue): First in First out

1. 隊伍有前方(front) 以及後方(back)
2. 進入隊伍(Push)，一定從back進入
3. 若要離開隊伍(Pop)，一定要從front離開

有N筆且已排序的資料，以二分搜尋法(binary searching)找尋一筆目標資料時，問最多"比對資料"幾次即可得知結果？

Ans : $\log_2 N + 1$

二分搜尋法

假如今天要搜尋的數列是有序的，我們便可以把線性搜尋法再做優化，使時間複

雜度再更低一點

二分搜尋法的原理跟小時候大家玩「終極密碼」的流程十分類似
就是那個 1~99 要你猜數字的遊戲

為了快點猜到（或是讓敵人快點猜到），有些人第一個數字會喊 50，為什麼呢？

因為無論數字是小於或是大於 50，剩下的數字一定會砍一半，變成原本的 1/2
假設下一次也繼續這樣砍對半，大概猜個七八次，就能「保證」一定猜得到

這邊可以做個簡單的小驗證：

如果只有 1 個數字，猜 1 次必定猜得到

如果只有 2 個數字，猜 2 次必定猜得到

如果只有 3 個數字，猜 2 次必定猜得到

如果只有 4 個數字，假設是 1 2 3 4 好了

切半猜 2，結果範圍變成 3 4，剩兩個數字，要猜 2 次

所以 4 個數字的話，猜 $1 + 2 = 3$ 次一定猜得到

如果有 8 個數字，切半剩下 4 個，所以要猜 $1 + 3 = 4$ 次

...

這樣繼續推廣下去，就會發現保證能猜到的次數與以 2 為底取 log 有關

詳細數學公式就不再贅述

圖 Ref : [Binary searching](#)

一記憶體位址範圍 6000(16) ~ 8FFF(16)，每一位址可儲存16bits，問該記憶體容量為多少KB?

Ans: (最大-最小)+1 個位址 * 16 / 8 Bytes / 1024 KB

8FFF (16)- 6000 (16) + 1 = 3000 (16) = 12288(10)

12288 * 16 / 8 / 2¹⁰ = 24KB

各式唯獨記憶體(ROM)

用電腦脈衝方式來"抹除"已儲存的資料 : EEPROM

ROM

英文全名為 [Read-Only Memory](#)，中文名稱叫做「唯讀型記憶體」。這種記憶體上的邏輯電路一開始都是設計規劃好的，資料都是固定的，不能做任何更改。因此ROM上的資料只能被讀取，不能做任何寫入的動作。

[PROM](#) (Programmable Read-Only Memory) 內部所含的電路熔絲可以被燒路為特定組態，資料一經寫入即永久存在，發展至今，也有產生幾種不同形式的ROM，包括了 [EPROM](#)、[EEPROM](#)以及[Flash ROM](#)。







EPROM

英文全名為 [Erasable Programmable Read-Only Memory](#)，中文名稱為「可抹除程式化唯讀記憶體」。這種ROM可以用紫外線燈來消除內部資料，以前主機板上的BIOS都是燒在這裡。

EEPROM

英文全名為 [Electrically Erasable Programmable Read-Only Memory](#)，中文名稱為「電子式可抹除程式化唯讀記憶體」。這種ROM不需要紫外線的照射，只需要用電氣的方法，在數秒內即可將ROM上的資料清除，然後再用類似燒錄的方法，把所需的資料放到ROM上。

圖 Ref : [ROM](#)

| | AND | OR | NAND | NOR | XOR | XNOR |
|-----|---|---|---|---|---|---|
| 輸入 | 輸出 | 輸出 | 輸出 | 輸出 | 輸出 | 輸出 |
| 0 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 符號 |  |  |  |  |  |  |

$$(x \text{ XOR } y) \text{ XOR } y' = A$$

$$A = x'$$

結合律: $A \text{ XOR } (B \text{ XOR } C) = (A \text{ XOR } B) \text{ XOR } C$

Ans: $x \text{ XOR } 1$

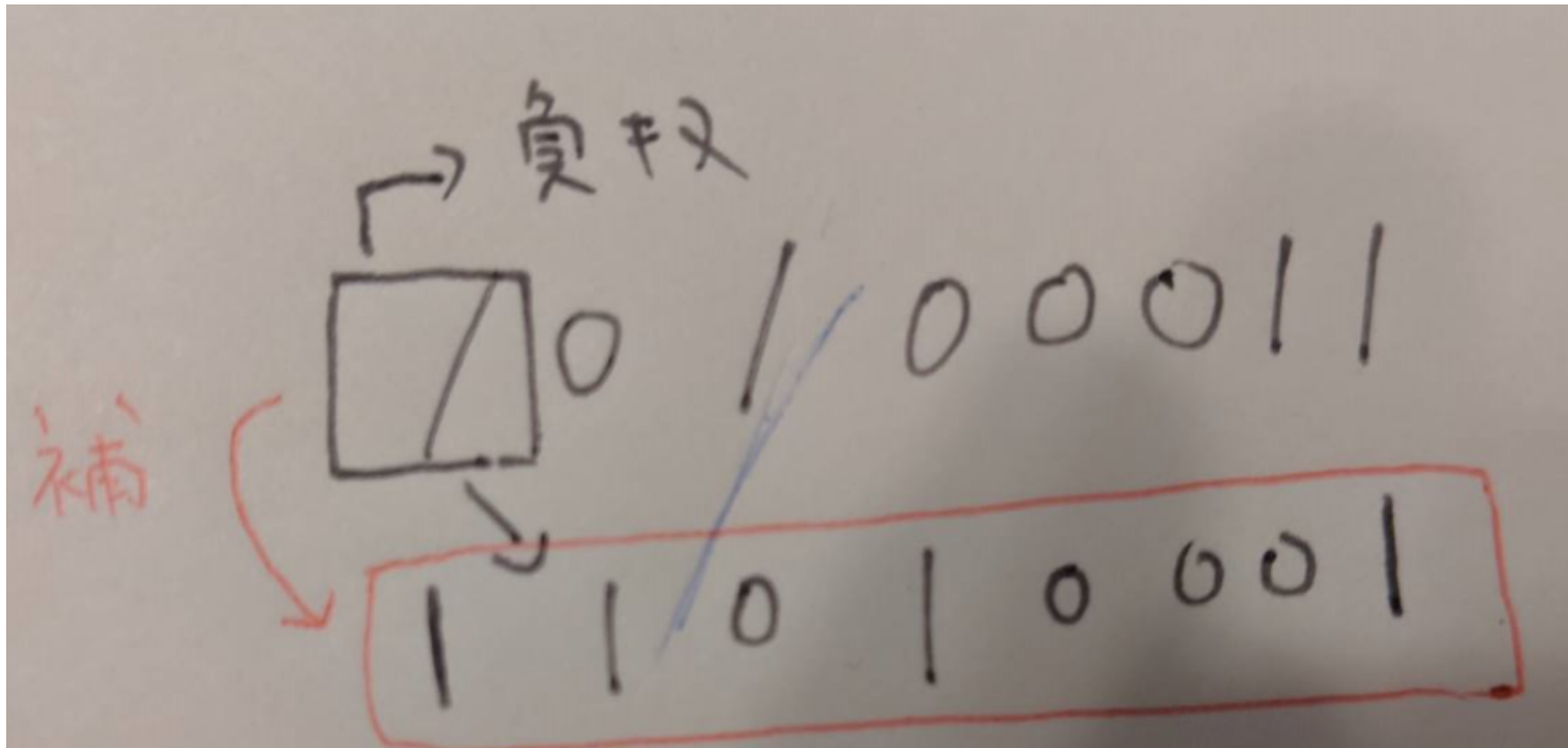
if $x == 1$ result is 0

if $x == 0$ result is 1

conclusion: x'

算數右位移(正數補0，負數補1)

ex: 10100011(2) 右移一位



補:

算術移位運算(Arithmetic shift operation)

- * 2 補數算術右移：每個位元向右移動一個位置，最左邊位元複製原位元值(維持符號) \rightarrow 相當於將原數值除以 2
- * 2 補數算術左移：每個位元向左移動一個位置，丟棄最左邊的位元，最右邊填入 0 \rightarrow 相當於將原數值乘以 2
- * 例如(2 補數)：
 - # 將 $1001\ 1001_2 (-103_{10})$ 算術右移 $\rightarrow 1100\ 1100_2 (-52_{10})$
 - # 將 $1101\ 1001_2 (-39_{10})$ 算術左移 $\rightarrow 1011\ 0010_2 (-78_{10})$
 - # 將 $1001\ 1001_2 (-103_{10})$ 算術左移 $\rightarrow 0011\ 0010_2 (+50_{10}, \text{溢位})$

22. 設 $A = 100011_{(2)}$ 、 $B = 011011_{(2)}$ 皆為 2's 補數形式 (長度 6 位元)，試問 $A + B = ?$ (A) $-2_{(10)}$ (B) $-1_{(10)}$ (C) $1_{(10)}$ (D) $2_{(10)}$ 。 100010

二補數 (英語: **2's complement**) 是一種用二進位表示有號數的方法，也是一種將數字的正負號變號的方式，常在電腦科學中使用。二補數以有符號位元的二進位數定義。

正數和 0 的二補數就是該數字本身。負數的二補數則是將其對應正數按位元取反再加 1。

A 為負數 $100011 - 1 = 100010 \rightarrow 011101 \Rightarrow -29$

B 為正數 $011011 \Rightarrow 1+2+8+16=27$

$-29+27=-2$

23. 設長度 10 位元的二進位數具 2's 補數形式，則該數值可表示的範圍為何？ (A) $-512_{(10)} \sim 512_{(10)}$ (B) $-512_{(10)} \sim 511_{(10)}$ (C) $-511_{(10)} \sim 512_{(10)}$

512₍₁₀₎ ~~(10)~~ - 511₍₁₀₎ ~ 511₍₁₀₎ °

2ⁿ - 1

011101

一考皆屬於“物件導向式”

符
號


| | | | | | | | | |
|---|---|---|---|---|---|---|---|--------|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | = 127 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | = 2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | = 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | = 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | = -1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | = -2 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | = -127 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | = -128 |


(C) LISP (D) COBOL °

(A) 28. 有關 MPEG 編碼 (encoding) 方法，其資料壓縮 (compression) 特性為下列何者？
(A) 有損式 (lossy) 視訊壓縮 (B) 有損式 (lossy) 圖像壓縮
(C) 無損式 (lossless) 視訊壓縮 (D) 無損式 (lossless) 圖像壓縮。

MP3 [編輯]

維基百科，自由的百科全書

 本文介紹的是一種音訊編碼格式。關於播放這種格式音訊的裝置，請見「[MP3播放器](#)」。

 關於與「**MP3**」名稱相近或相同的條目，請見「[MP3 \(消歧義\)](#)」。

動態圖像專家組-1或**動態圖像專家組-2 音訊層III**（英語：**MPEG-1 or MPEG-2 Audio Layer III**），常簡稱為**MP3**，是當今流行的一種數位音訊編碼和失真壓縮格式，它被設計來大幅降低音訊資料量，通過捨棄PCM音訊資料中對人類聽覺不重要的部分，達成壓縮成較小檔案的目的。而對於大多數用戶的聽覺感受來說，MP3的音質與最初的不壓縮音訊相比沒有明顯的下降。它是在1991年，由位於德國埃爾朗根的研究組織**弗勞恩霍夫協會**的一組工程師發明和標準化的。MP3的普及，曾對音樂產業造成衝擊與影響。

MPEG正式稽核程式是**Moving Picture Experts Group**的簡稱。這個名字本來的含義是指一個研究影片和音訊編碼標準的「動態圖像專家組」組織，成立於1988年，致力開發影片、音訊的壓縮編碼技術。現在我們所說的MPEG泛指由該小組制定的一系列影片編碼標準正式稽核程式。該小組於1988年組成，至今已經制定了MPEG-1、MPEG-2、MPEG-3、MPEG-4、MPEG-7等多個標準，MPEG-21正在制定中。MPEG是ISO和IEC的工作群組，它的官方頭銜為：第一技術委員會第二十九子委員會第十一號工作群組正式稽核程式，英文頭銜為ISO/IEC JTC1/SC29 WG11。MPEG大約每2-3個月舉行一次會議，每次會議大約持續5天，在會議期間，新的建議和技術細節先在小組中討論，成熟後進入標準化的正式稽核程式。與MPEG工作群組相關的其他幾個影片標準化工作群組包括ITU-T VCEG以及JVT。

- (C) 29. 設計高品質的軟體模組時，應力求何種特性 (cohesion) 為何？ (A) 低耦合性、低內聚性 (B) 高耦合性、低內聚性 (C) 低耦合性、高內聚性 (D) 高耦合性、高內聚性。
- A) 30. 軟體開發程序所用之“瀑布” (waterfall) 模型，其進行的四大步驟依序為何？ (A) 分析→設計→實作→測試 (B) 分析→設計→測試→實作 (C) 設計→分析→實作→測試 (D) 設計→分析→測試→實作。

HyperText Transfer Protocol

貳、非選擇題二大題 (每大題 20 分)

一、(一) 請問 ADSL 與 HTTP 兩者之英文 (或中文) 全名為何？ (4 分)

(二) 邏輯運算式如右所示 $(7A7_{(16)} \text{ XOR } B8B_{(16)}) \text{ OR } 123_{(16)}$ ，其中各運算元長度皆為 12 位元，XOR 表“互斥”運算，OR 表“或”運算，請將計算結果以 16 進制式作表示？ (6 分)

(三) 請說明於電腦內建置“快取” (cache) 記憶體之主要目的為何？ (4 分)

OS 四一群程序 (processes) 於電腦內發生“死結” (deadlock) 所需的四項要件中，除了“互斥” (mutual exclusion) 一項外，其餘三項為何？ (4 分)

二、(一) 請說明多重程式規劃 (multiprogramming) 方法之主要特性為何？

(二) 以需求分段法 (demand segmentation) 用於配置記憶體過程中，易引起“外部碎裂” (external fragmentation) 的現象，試問造成“碎裂”的原因為何？

(三) 1. 請說明以分頁法 (paging) 用於配置記憶體之主要特點為何？

2. 於分頁法中，當頁 (page) 的分割長度“過大”或者“過小”之情況下，會有何影響？請分別說明之。

Ans 29(C) 30(A)

一、(一) ADSL (Asymmetric Digital Subscriber Line)：非對稱數位用戶線路。

HTTP (HyperText Transfer Protocol)：超文本傳輸協定。

$$(二) 7A7_{(16)} = 011110100111_{(2)}$$

$$B8B_{(16)} = 101110001011_{(2)}$$

$$123_{(16)} = 000100100011_{(2)}$$

$$\rightarrow 011110100111 \oplus 101110001011 = 110000101100$$

$$\rightarrow 110000101100 + 000100100011 = 110100101111 = D2F_{(16)}$$

三當 CPU 在處理資料時，它會先到 cache 當中去尋找，如果 CPU 要處理的資料先前已經存在過 cache 中的話，就不用去主記憶體 (main memory) 去尋找。因為 cache memory 的存取速度較主記憶體快上許多，因此可以加快處理的速度。

四 Hold & Wait (持有並等待)

No preemption (不可搶奪)

Circular Waiting (循環等待)

二、(一)系統或記憶體允許多個 processes 同時執行，當執行中的 process 因為某種原因無法繼續往下執行時 (wait for I/O、wait for resource 等)，可將 CPU 切換給其他 processes 使用，這樣可以使得 CPU 盡可能的忙碌，以發揮它的效能。

二在連續性配置策略底下，若任何一個 Free Block 皆無法滿足 process 的要求，但是這些 Free Block 加起來的總和大於 process 的要求。但是因為這些 Free Block 並非連續的，所以無法配置，造成記憶體的浪費。

三1.採取非連續性的配置策略，若 process 的大小為 N 個 pages，則 OS 只要在實體記憶體中找到大於等於 N 個 Free Frames (可以不連續) 就可以配置，避免了外部碎裂。

2.

| | | |
|------------------|-------------|-------------|
| | 分割過大 | 分割過小 |
| | 愈小 | 愈大 |
| page fault ratio | I/O time 減少 | I/O time 增加 |

| | | |
|----------|--------------------------|--------------------------|
| I/O time | (因為 page fault ratio 下降) | (因為 page fault ratio 上升) |
| 缺點 | 一次會占用太多的實體記憶體 | 造成 Thrashing 現象 |

七、十六進位與二進位的轉換

十六進位→二進位

將每一個十六進位數字都轉成二進位數字。

範例 $(5BD1.B)_{16} = (101101111010001.1011)_2$

$$(5BD1.B)_{16} = (\underline{0101} \ \underline{1011} \ \underline{1101} \ \underline{0001} . \underline{1011})_2$$

$$= (\ 5 \quad B \quad D \quad 1 \quad . \quad B)_{16}$$

二進位→十六進位

以小數點為基準，分向左右兩邊每4位取一組，不足4位則補0。

範例 $(111010011.101)_2 = (1D3.A)_{16}$

$$(\underline{0001} \ \underline{1101} \ \underline{0011} . \underline{1010})_2$$

$$= (\ 1 \quad D \quad 3 \quad . \quad A)_{16}$$

Fragmentation

外部碎裂 (External Fragmentation)

系統中，所有可用空間總和大於某個 process 所需要，但因為這些空間不連續所以無法配給該 process 使用，造成 memory 空間閒置。

解決方法：

Compaction：類似磁碟重組的概念，移動執行中的 process，使不連續的 free blocks 得以聚集成一塊夠大的連續可用空間
很難在短時間內決定一個最佳的壓縮策略。

process 必須是 dynamic binding 才可以支援。

Page memory management

內部碎裂 (Internal Fragmentation)

作業系統配置給 process 的 memory 空間大於 process 真正所需的空間，這些多出來的空間該 process 用不到，而且也沒辦法供其他 process 使用，形成浪費。

Reducing the page size can alleviate Internal Fragmentation.

Enlarging the page size helps to reduce the size of the page table.

[memory management](#)

